

Les boucles (d'or)

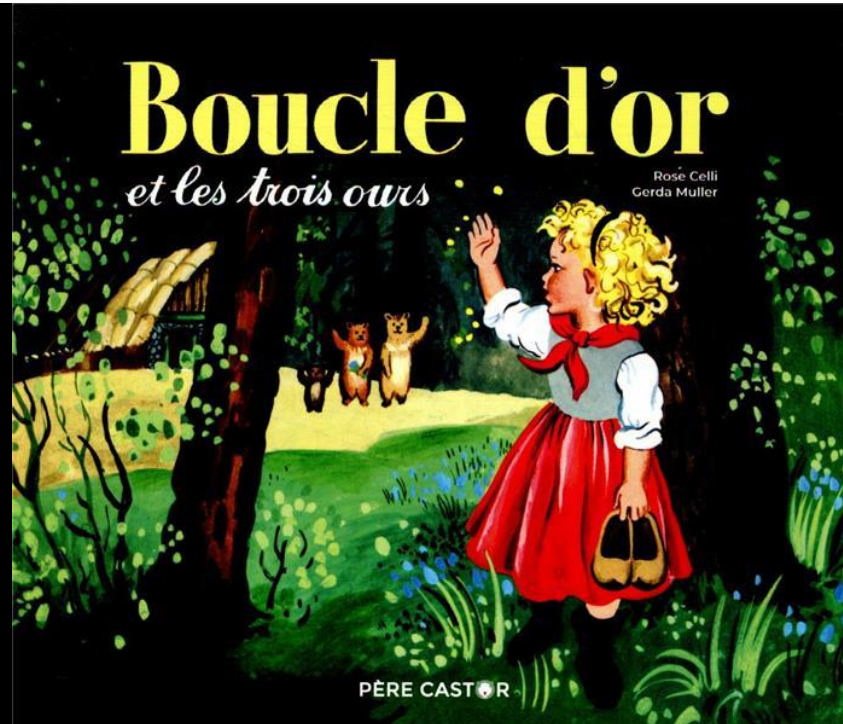
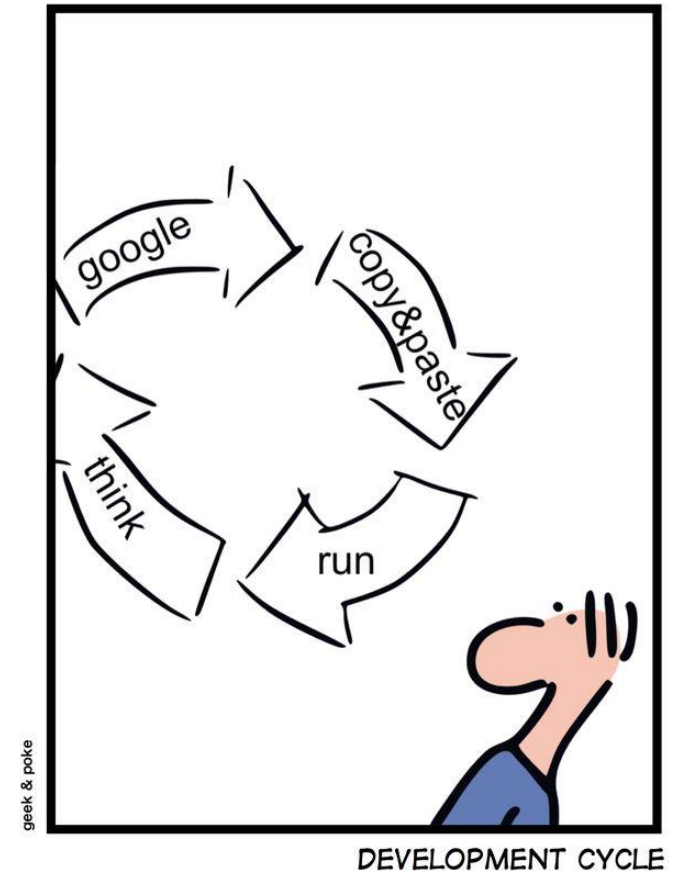


Table des Matières

- Résumé de l'épisode précédent
- Répéter des instructions
- La boucle pour (for)
- La boucle tant que (while)
- La boucle répéter jusqu'à (do while)

SIMPLY EXPLAINED



Dans l'épisode précédent...



- `if (condition) {`
 `...`
} `else if (condition2) {`
 `...`
} `else { ... }`
- `==` `!=` `<` `<=` `=>` `>`
- `&&` `||` `!`
- `switch (expression) {`
 `case valeur1:`
 `...`
 `break;`
 `default:`
 `...`
}

Répéter des instructions

- Essayons d'écrire un programme qui affiche tous les nombres entre 1 et 5 (avec nos connaissances actuelles)

```
console.log(1);  
console.log(2);  
console.log(3);  
console.log(4);  
console.log(5);
```

- Exercice : faites la même chose pour les nombres de 1 à 1000...

Répéter des instructions

- Il est possible de répéter l'exécution d'un ensemble d'instructions en plaçant ces instructions à l'intérieur d'une **boucle**
- Le nombre de répétitions peut être connu à l'avance ou dépendre de l'évaluation d'une condition.
- On appelle un tour de boucle une **itération**

La boucle pour (for)

- On a fréquemment besoin d'écrire des boucles dont la condition est basée sur la valeur d'une variable qui est modifiée dans le corps de la boucle
- L'instruction **for** répond à ce besoin

```
for (let compteur = 1; compteur <= 5; compteur++) {  
    console.log(compteur);  
}
```

La boucle pour (for)

```
for (initialisation; condition; étape) {  
    // instructions exécutées tant que la condition est vérifiée  
}
```

- **L'initialisation** se produit une seule fois, au début de l'exécution.
- La **condition** est évaluée avant chaque tour. Si elle est vraie, un nouveau tour de boucle est effectué. Sinon, la boucle est terminée.
- **L'étape** ou **incrément** est réalisée après chaque tour de boucle.

La boucle pour (for) – cas d'erreur possibles

```
for (let i = 1; i <= 5; i++) {  
    console.log(i); // OK  
}  
console.log(i); // Erreur : la variable i n'est pas visible ici
```

```
for (let i = 1; i <= 5; i++) {  
    console.log(i);  
    i++; // La variable est modifiée dans le corps de la boucle  
}
```


La boucle tant que (while)

- La boucle **while** permet de répéter des instructions **tant qu'une condition est vérifiée**.

```
let nombre = 1;
while (nombre <= 5) {
  console.log(nombre);
  nombre++;
}
```

La boucle tant que (while)

```
while (condition) {  
    // instructions exécutées tant que la condition est vérifiée  
}
```

- Avant chaque tour de boucle, la condition associée au **while** est évaluée
 - Si elle est vraie
 - Les instructions du bloc de code associé au **while** sont exécutées.
 - Ensuite, l'exécution revient au niveau du **while** et la condition est à nouveau vérifiée.
 - Si elle est fausse
 - Les instructions du bloc ne sont pas exécutées
 - Le programme continue juste après le bloc **while**.

La boucle tant que (while) – attention !

```
let nombre = 1;
while (nombre != 5) {
  console.log(nombre);
  nombre+=3;
} // Boucle infinie car la condition est trop précise
```

```
while (true) {
  // Parfois, on a besoin d'avoir une boucle qui continue
  // encore et encore...
}
```

La boucle répéter jusqu'à (do while)

- La boucle répéter jusqu'à (**do-while**) est presque identique à la boucle tant que (**while**)
- Les instructions contenues dans le bloc **do** sont toujours exécutées au minimum une fois, même si la condition est fausse

```
let nombre = 1;  
do {  
    console.log(nombre);  
    nombre++;  
} while (nombre != 5);
```

Choix entre while et for

- Si on peut prévoir à l'avance le nombre de tours à effectuer, on utilisera de préférence un **for** car sa syntaxe intègre la modification du compteur
 - Parcours d'une liste
 - Nombre de tour donné par une variable
- Sinon, on utilisera un **while** car toutes les boucles peuvent s'écrire avec un **while**

```
let lettre = "";  
while (lettre !== "X") {  
    lettre = prompt("Tapez une lettre ou X pour sortir : ");  
    // ...  
}
```



Exercices

- Exercices 5.x : **Boucles**

NOM :				😊		😐		😞	
Date :				lundi	mardi	mercredi	jeudi	vendredi	
septembre	octobre	novembre	décembre	janvier	février	mars	avril	mai	juin
Objectif : Tracer des boucles montantes.									



Repasse sur les lignes pointillées pour faire de belles boucles.

